Master's thesis
Master's Programme in Data Science

# (Re)lexicalization of auto-written news with contextual and cross-lingual word embeddings

Miia Rämö

October 22, 2020

Supervisor(s): Prof. Hannu Toivonen
Leo Leppänen, M.Sc.

Examiner(s): Prof. Hannu Toivonen
Mark Granroth-Wilding, PhD

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

| Tiedekunta — Fakultet — Faculty | Koulutusohjelma — Utbildningsprogram — Degree programme |
|---|---|
| Faculty of Science | Master's Programme in Data Science |

Tekijä — Författare — Author

Miia Rämö

Työn nimi — Arbetets titel — Title

(Re)lexicalization of auto-written news with contextual and cross-lingual word embeddings

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidantal — Number of pages |
|---|---|---|
| Master's thesis | October 22, 2020 | 45 |

Tiivistelmä — Referat — Abstract

In news agencies, there is a growing interest towards automated journalism. Majority of the systems applied are template- or rule-based, as they are expected to produce accurate and fluent output transparently.

However, this approach often leads to output that lacks variety. To overcome this issue, I propose two approaches. In the lexicalization approach new words are included in the sentences, and in relexicalization approach some existing words are replaced with synonyms. Both of the approaches utilize contextual word embeddings for finding suitable words.

Furthermore, the above approaches require linguistic resources, which are only available for high-resource languages. Thus, I present variants of the (re)lexicalization approaches that allow their utilization for low-resource languages. These variants utilize cross-lingual word embeddings to access linguistic resources of a high-resource language.

The high-resource variants achieved promising results. However, the sampling of words should be further enhanced to improve reliability. The low-resource variants did show some promising results, but the quality suffered from complex morphology of the example language. This is a clear next issue to address and resolving it is expected to significantly improve the results.

ACM Computing Classification System (CCS):
Computing methodologies → Artificial intelligence → Natural language processing → Natural language generation

Avainsanat — Nyckelord — Keywords

natural language generation, word embeddings, automated journalism

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

# Contents

# 1. Introduction

News agencies around the globe are interested in generating news reports based on the vast amounts of data available automatically to provide news faster and cheaper – to get advantage on other companies. The automatically generated news reports can be used either as raw material by journalists, or as finalized reports on e.g. weather or sports. The former allows journalists to concentrate their time on analyzing the deeper causes and the final story-telling, to produce higher quality news reports. The latter enables news agencies to offer stories that are of interest to numerous marginal groups of people as a automated news generation system can produce massive amount of reports in nearly real-time. For example, if a system is implemented to report some sport, the same system can generate news about all matches, be they professional or amateur series matches, assuming the data is available in the correct format. Another interesting possibility is personalising content to the specific preference of each reader.

Automated journalism is achieved with *natural language generation* (NLG) and according to Leppänen et al. [23] it has the following six requirements: transparency, accuracy, modifiability and transferability of the system, fluency of the output, data availability and topicality of news. In general, there are two approaches: hand-crafted and trainable systems [14], where the former has wider acceptance in the news industry [41]. Systems that rely on hand-crafted rules and templates fulfil the requirement of transparency, since every decision can be traced back, while trainable systems are often black boxes, whose generation process cannot be supervised. However, human-crafted systems have various downsides: they are expensive and laborious to implement and their transferability is poor, since they are domain specific. A major downside of these hand-crafted systems is that their output often lack variety.

In this thesis, I propose an NLG system extension for additional variety. The proposed extension is two-part: in *lexicalization* approach some new words are chosen to be added to the NLG output, and in *relexicalization* approach some words in the original sentence are replaced with synonyms. The lexicalizations are retrieved using a contextual word embedding model and further filtered based on parts-of-speech, and for relexicalizations the potential synonyms are retrieved from a synonym dictionary. Suitability of these lexicalizations and relexicalizations is evaluated for the context using a contextual

word embedding model.

The solution described above requires that the language to be *(re)lexicalized* has a part-of-speech tagger and a synonym dictionary available. Therefore the proposed approaches on their own are insufficient for low-resource languages. In this thesis I further propose a solution where cross-lingual word embeddings are utilised to find a high-resource language equivalents for low-resource language words. This allows usage of wider language resources with more languages.

Rest of this thesis is organized as follows. In Chapter 2, I present the relevant background, i.e. NLG, automated journalism, word embeddings and evaluation of NLG, and context, for the thesis topic. In Chapter 3, I describe my research design and objectives. Then, in Chapter 4, I present the algorithms developed during this thesis work. In Chapter 5, I describe the evaluation setup, and present and discuss the results from human evaluation conducted. Finally, in Chapter 6 I conclude this thesis and identify some future directions.

# 2. Background

In this chapter, I introduce the relevant background required as context for my study. In Section 2.1 I give a brief introduction to Natural Language Generation (NLG) to motivate the need for additional variety. In Section 2.2, I introduce the field of automated journalism, which is the more specific domain of interest. In Section 2.3, I introduce a technique widely used in modern NLP – word embeddings. Finally, the outputs produced by my NLG algorithms have to be evaluated to measure if the modification are a success. The relevant background on evaluation of NLG is introduced in Section 2.4.

## 2.1 Natural language generation

*Natural language generation* (NLG) is defined by Reiter & Dale [37, p. 1] as "the sub-field of artificial intelligence and computational linguistics that is concerned with the construction of computer systems than can produce understandable texts in English or other human languages from some underlying non-linguistic representation of information."

NLG systems always aim to produce similar type of output – text. Meanwhile, the input of systems can be quite different [14]. The most common type of NLG systems is *data-to-text* systems, where the input is structured data. Some other types are *text-to-text* systems, where the input is some other text [50], and *vision-to-text* systems, where computer vision input is described with text [43].

Some concrete examples of data-to-text natural language generation systems are those that generate soccer reports [6], financial reports [32], weather forecasts [35], patient information summaries [2], personalised environmental information [49], and letters to persuade readers to quit smoking [38].

The whole end-to-end process of NLG is often conceptualized in terms of six distinct sub-tasks [14, 37]. However, there are other approaches for this division. The sub-tasks, together with the questions they answer, are the following:

1. **Content determination** – What information will be included in the final report?
2. **Discourse planning** – In which order will the content be presented?

3. **Sentence aggregation** – How is the chosen information divided into sentences?
4. **Lexicalisation** – What are the right words and phrases to express the information?
5. **Referring expression generation** – What are the words and phrases used to refer to domain objects and people?
6. **Linguistic realisation** – Finally, words and phrases chosen in above tasks are combined to form the final, well-written report.

The listed tasks are organised in different systems in various ways. These different approaches can be divided into three categories [14]: Modular pipelines, Planning-based approaches, and Integrated or global approaches. The modular pipelines are the traditional and most well-known approach where the NLG system consists of modules that perform one or multiple tasks of those introduced above. In planning-based approaches text generation is viewed as execution of actions that lead to new states in a planned manner. The current trend are integrated or global approaches, where systems are heavily based on statistical learning. The architectural approach can be either end-to-end, where the division to tasks of NLG is blurred, or a neural pipeline, where there are intermediate representations present as in traditional pipeline, but state-of-the-art deep learning methods are used [11].

In the task of linguistic realisation, the order in which parts of a sentence are presented is decided upon. Linguistic realisation may be done in various ways, but Gatt and Krahmer [14] discuss the following three: human-crafted templates, hand-coded grammar-based systems and statistical approaches.

In hand-coded grammar based approaches the decisions related to language realisation are made based on the grammar of the language in question. In other words, the rules of writing and speaking the language, which people learn gradually as they learn a language as a child. To mention why these systems are problematic: 1) hand-coding these complex rules is naturally very laborious, and 2) the systems often require a very detailed input.

Template-based approaches are applicable when the data in questions can be described with a relatively small number of sentences. I present an example of templates and output of a hypothetical template-based weather reporting system in Figure 2.1. The part above the line would be the template where some values are embedded, and the part below the line would be the final output.

Using hand-crafted grammar rules or templates is expensive since constructing them is time consuming. Although the templates are often simple, a large number of them is required to achieve varied output. Moreover, these templates need to be implemented again for each new domain. However, these systems produce output that is grammatically

On **$date** the weather was **$weather_description**.
The average temperature was **$avg_temperature** °C.
The average wind speed was **$avg_wind** m/s.
The amount of rainfall for that day was **$rainfall** mm.

---

On **01.05.2020** the weather was **sunny**.
The average temperature was **8** °C.
The average wind speed was **5** m/s.
The amount of rainfall for that day was **0** mm.

**Figure 2.1:** A simple example of how templates are utilized to generate fluent NLG output.

and factually correct [14]. Thus, they respect the requirements of fluency and accuracy of output for automated journalism. The output is however often very stiff, while trainable systems often generate more varied text.

Due to the problems with systems that require heavy human labour, it is natural that the emergence of statistical learning methods has shifted the interest towards the integrated approach, and furthermore towards neural pipeline approaches [11]. These approaches however require significant amount of data to train on and are thus sub-optimal solutions when working with low-resource languages, where there is a lack of quality data.

Another problem with the integrated approaches is that they are often black boxes, which is an issue especially for generation of news texts, while a black box solution does not respect the requirement of transparency. This will be discussed more thoroughly in the following section.

## 2.2 Automated journalism

In automated journalism, or robojournalism, some data is transformed into human readable news reports with natural language generation techniques introduced earlier. Graefe [17, p.9] explains the motivation for automated journalism as follows: "... not only can algorithms create thousands of news stories for a particular topic, they also do it more quickly, cheaply, and potentially with fewer errors than any human journalist." According to them, automated journalism is the best fit when there is structured data available, and the topic is repetitive in nature.

According to Graefe, there are multiple potential benefits of automated journalism. Firstly, automatization allows speedy publishing of news reports – a report can be published nearly real-time when the source data becomes available. Secondly, the scale in

which reports are published can be increased – for example, rather than reporting only the major earthquakes, a report can be published about all observations by seismographic sensors without running out of journalistic resources. Thirdly, automated journalism systems are argued to be less error-prone as they do not make mistakes such as misspelling or calculation errors. In other words, their accuracy is higher than that of human journalists' – assuming that there is no errors in the code and that the source data is correct. Furthermore, again assuming that no subjectivity is coded into the system and the data is objective, the system will produce objective output. Finally, automatization allows personalization of news report for smaller target groups – even individuals, and offering news on demand.

In addition to the potential benefits, Graefe mentioned the following limitations. Firstly, an automated journalism system can only be as good as the data it uses. In other words, the availability and quality of the source data is key to success. Secondly, although the system might identify interesting events in the source data, it cannot ask the question "why?". Thus, human validation and reasoning is still required. Thirdly, according to Graefe, algorithms lack ingenuity, and are thus limited in their ability to observe society and fulfill journalistic tasks. Finally, people prefer reading human-written rather than automated news, according to experimental evidence. It should be noted that NLG has advanced tremendously after this article was published in 2016.

Leppänen et al. [23] identified six specific requirements for automated journalism: transparency, accuracy, modifiability and transferability of the system, fluency of output, data availability, and topicality of news. These requirements were derived from those identified for human journalists.

Graefe further explains that the process of automated journalism has five phases: collection of data, identification of interesting events, prioritization of insights, generation of the narrative and finally publishing the story (Figure 2.2). They emphasise that automated journalism systems require input from domain experts to define domain specific rules, and criteria of newsworthiness.

Graefe gives an overview of the status of automated journalism in newsrooms in 2016 [17]. The used systems are developed either in-house or by an external party specialized in offering NLG solutions. Examples of in-house solutions are systems that automate homicide and earthquake reporting for Los Angeles Times. The developer of these systems, Ken Schwencke, described them as "embarrassingly simple", as they are template-based. In other words, they use pre-written pieces of text and insert numbers from the data into them (Section 2.1). As another example, one of the world's major news organizations, Associated Press, began reporting quarterly company earnings automatically in 2014 using Wordsmith platform by Automated Insights.

In their study, Sirén-Heikel et al. [41] put together 26 interviews with American


**Figure 2.2:** The process of automated journalism as described by Graefe [17].

and European media representatives. Interviews were done between 2015 and 2018. 13 out of the 26 interviewees mentioned domain-specific, template-based NLG as the type of automation used at their media house. Thus, it is safe to say that the transparent approaches with template-based systems are appreciated by the industry.

## 2.3 Word embeddings

Interestingly, J. R. Firth summarized as early as in 1957 that "you shall know a word by the company it keeps" [12, p. 11]. In other words, if the meaning of a word is unknown, it can be figured out by the context the word appeared in.

This idea has later led to the emergence of word embeddings – distributed representations of words in a vector space. One of the earliest implementations was proposed in 1986 by Rumelhart et al. [40]. Word embeddings gained wide interest in 2013 when Mikolov et al. [26] proposed an efficient method for learning high-quality word vectors with unstructured textual data.

What makes word vectors especially interesting, is that they are able to capture semantic and syntactic relationships between words [28]. Furthermore, these relationships can be represented as linear translations. One regularly used (best scenario) example of operating with word vectors is the following [29]:

$$vec(king) - vec(man) + vec(woman) = vec(queen)$$

Here, the vector representing the word man is reduced from the vector of the word king. Adding the vector representing the word woman results in the final word being the word queen. In other words, the "maleness" of the word was replaced with "femaleness".

A division of word embedding models to two categories can be done based on whether a model assigns multiple vectors for a spelling based on the context, or whether a single vector is assigned based on all contexts the word appears in. The former are referred to as contextual and the latter as non-contextual word embeddings.

### 2.3.1 Non-contextual word embeddings

A well known technique to learn non-contextual word embedding is Word2Vec. This technique takes advantage of the idea of J. R. Firth. That is, words that appear in similar context are presumably similar and have similar vector representations. Word2Vec involves two methods for learning word embeddings, both of them proposed by Mikolov et al. [26]: continuous bag-of-words model (CBOW) and continuous skip-gram model. Simply put, the CBOW model is able to predict a word based on the surrounding context, and the Skip-gram model vice versa is able to predict the context for a given word (Figure 2.3).

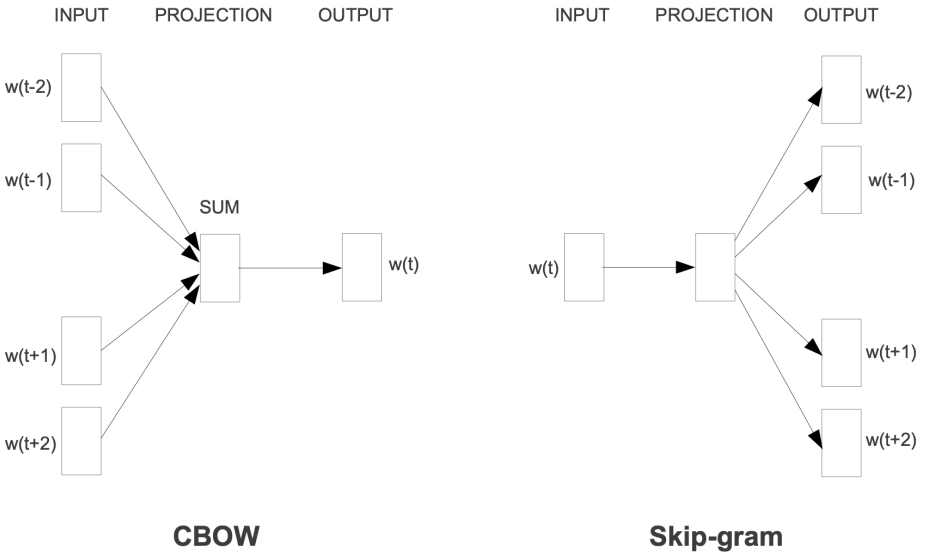


**Figure 2.3:** Methods for learning word embeddings involved in Word2Vec [26]. The continuous bag-of-words model (CBOW) on the left is able to predict a word based on the surrounding context. The skip-gram model on the right is able to predict the surrounding context given a word.

Words that are not present in the training data are called out of vocabulary – OOV – words. Especially when working with languages that have a complex morphology, encountering these words when using the model is common. This is due to one word having multiple morphological forms, some of which are quite rare. An example tuplet of a com-

mon and an uncommon morphological forms in Finnish is *(kaupan, kauppansakinkohan)*. Both of the words are morphological forms of the Finnish word 'kauppa' which translates to English as 'shop'. As Word2Vec creates vectors out of whole words, it is not able to handle these OOV words. Another non-contextual word embedding model, FastText, has a solution for this problem.

FastText [4] is an extension to the model proposed by Mikolov et al. [27]. Where Word2Vec, as described above, assigns a distinct vector for each word in a corpus disregarding the morphology, in FastText each word is represented as a bag of character $n$-grams, or in other words, a bag where the word is split into varying length parts. Those $n$-grams are each associated with a vector and the complete word is represented as a sum of these $n$-gram vectors. This ability to take into consideration the internal structure of words enables the handling of OOV words, when those words can be constructed from character $n$-grams present in the corpus.

Another issue, on top of the one with OOV words, is presented by polysemous or homonymous words. Being able to define whether a word is polysemous or homonymous is unnecessary for the purposes of this thesis, but to briefly put it: a word is polysemous, if it is used to describe different things and, a set of words are homonymous if they have the same spelling or pronunciation but are nevertheless different words. An often used example of homonymy is the word *cell* - in one context it could mean a cell of a living organism and in another it could mean a prison cell.

As both Word2Vec and FastText define a distinct vector for each word, where a word is a sequence of characters with the same spelling, they lack the ability to tell apart the different meanings of polysemous and homonymous words. In other words, the possible multiple meanings of words are expected to be represented by one vector.

### 2.3.2 Contextual word embeddings

Contextual word embeddings models have been introduced to solve the problem of words having multiple senses [36, 21]. In the case of contextual word embeddings, there is a separate vector in the vector space for each context a word appears in, context is here typically defined as the surrounding sentence. That is, a homonym *bark* would not be condensed into one single vector but rather separate vectors would represent its meaning as the bark of a tree and the bark of a dog. To further clarify, in case of non-contextual word embeddings, there is one constant vector representation for a word, but in case of contextual word embeddings, the representation is no longer constant.

Some examples of contextual word embedding models are ELMo by AllenNLP [31], BERT by Google Research [9], GPT, GPT-2 and GPT-3 by OpenAI [33, 34, 5], and variations of BERT such as RoBERTa by Facebook AI [25].

Whereas non-contextual word embedding models introduced above are simple feed-forward neural networks, contextual word embedding models are more complex. The architecture of ELMo for example is a bidirectional language model biLM, where both the forward and backward LM architectures are Long Short-Term Memories (LSTM) [20].

Whereas ELMo uses unidirectional language models to learn general language representations forward and backward, a state-of-the-art model BERT uses a new *masked language model* (MLM) pre-training objective [9]. This pre-training objective is inspired by the Cloze task [42]. In this MLM approach, some words in the input sentences are replaced with `[MASK]` tokens and the objective is, only with the context as information, to predict, what was the original word. Additionally, there is a second pre-training objective of *next sentence prediction* to understand relationships between sentences for tasks such as question answering. BERT's model architecture is derived from work by Vaswani et al. [47], and its name is descriptively an abbreviation of the words Bidirectional Encoder Representation from Transformers [9]. In other words, BERT contains multiple layers of encoders derived from the transformer architecture.

### 2.3.3 Cross-lingual word embeddings

In addition to making word embeddings contextual, recent works have explored making them cross-lingual. Ruder et al. [39] mention two reasons why cross lingual embeddings are interesting: they enable 1) comparison of meanings of words across languages, and 2) model transfer between languages. The former is key for example to bilingual lexicon induction and machine translation. The latter enables us to develop models in resource-rich languages and then transform them to low-resource languages.

Among the best known works on cross-lingual word embeddings is that of Mikolov et al. [27]. In their work, the authors train word embeddings for two languages separately and then align the spaces using a small set of parallel data – a dictionary of translations that define places where the spaces are known to overlap. While effective, the requirement for this dictionary is a restriction on the applicability of the method. Conneau et al. [8], in turn, show how the alignment can be achieved without parallel data – in an unsupervised fashion. These approaches enable mapping words between two languages, which further allows translation of these words between the languages. To elaborate, when languages are aligned, vectors representing the same word would be in roughly the same position in the vector space (Figure 2.4). Thus, retrieving the closest vector for a source word from a target language, that is aligned with the source language, provides a translation.

A toy illustration by Conneau et al. is presented in Figure 2.4. Part A of the figure represents how the monolingual vector sets are similar in shape but unaligned. The latter parts represent how the vector sets are transformed to align with each other.

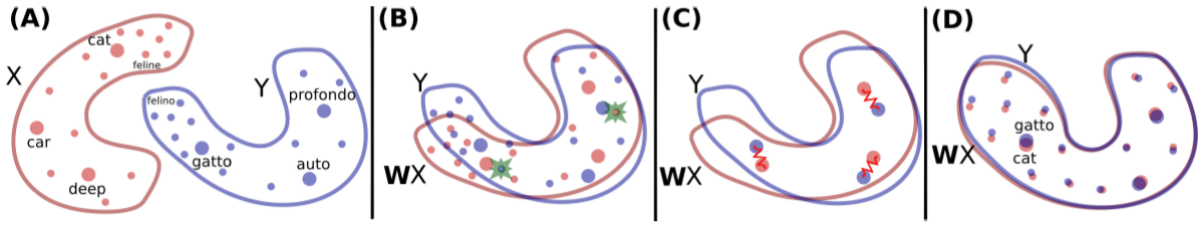


**Figure 2.4:** Toy illustration by Conneau et al. [8] showcasing how monolingual vector spaces are similar in shape but different in position of the words (A), and how the vector spaces can be aligned to map words between languages (B, C, D).

Mikolov et al. [27] show that cross-lingual word embedding achieve good results in translation task even between languages that are distantly related, such as English and Czech. Cross-lingual embeddings are thus a promising approach for solving the issues of NLP that low-resource languages face.

There are differences in type of alignment of data in approaches to cross-lingual embeddings. Alignment can be done at the level of words, sentences, or documents. Ruder et al. [39] provide an extensive survey of different approaches to cross-lingual embeddings.

Majority of research in field of cross-lingual embeddings is done in a bilingual setting. However, recent work [10, 24] shows that multilingual setting improves results reported on standard tasks such as word similarity prediction.

I described before the issues with polysemy and homonymy with monolingual word embeddings. According to Ruder et al. [39] the issue is amplified in cross-lingual word embeddings. They state that if we have $m$ bad word embeddings in the source language, and $n$ bad word embeddings in the target language, we can derive $O(nm)$ false nearest neighbors from our cross-lingual embeddings.

There are recent studies showing promising results on tackling the issue of ambiguous words in bilingual [7] and multilingual [45] settings.

## 2.4 Evaluation of NLG

Evaluation is an important part of an NLG system, as it helps to understand e.g. whether the system successfully achieves the desired level of quality, or whether the system output had the desired impact [15].

Evaluation of NLG can be either intrinsic or extrinsic. An intrinsic evaluation measures system performance on its own, while extrinsic evaluation measures system performance in achieving a desired goal [14]. Hastie & Belz [19] broke these categories further down. For intrinsic evaluation they identified subcategories of output quality measures and user like measures. For extrinsic evaluation they identified subcategories of user task success and system purpose success. To further clarify the difference, I will describe

evaluation settings by Law et al. [22]. Their aim was to compare how textual and graphical representations of time series data performed in supporting medical decision making. Here, the intrinsic evaluation setup is the one where subjects are asked whether they prefer graphical or textual representations, whereas the extrinsic evaluation is observed in action. That is, quality of their decisions was measured when being presented textual versus graphical representations.

Intrinsic evaluation is more popular, as extrinsic evaluation is considered expensive and time consuming to conduct. Gkatzia and Mahamood [15] found that between years 2005 and 2015, 74.7% (59 out of 79 papers) of evaluation was intrinsic while only 15% (12 out of 79 papers) was extrinsic. The final 10.1% (8 out of 79 papers) reported both types of evaluation.

Automatic evaluation metrics are utilised for the output quality measures of intrinsic evaluation. The advantages of automatic evaluation metrics are the following: 1) evaluation is less subjective and thus more comparable between systems, and 2) conducting evaluation with automatic metrics is faster and cheaper. However, there are two main points for criticism [46]. Firstly, automatic metrics are uninterpretable as the scores require error analysis to specify the cause. Secondly, automatic metrics do not correlate with human evaluations.

User like measures, or human evaluation, can be conducted with few experts or with a reader-focused approach. The latter is often done by crowd sourcing to achieve high enough number of judges. Vougiouklis et al. [48] show that crowd sourcing and expert evaluation do not fully correlate. There are also discussion about possible issues with crowd sourcing. Fort et al. [13] discuss the ethical issues with crowd sourcing: 1) the people taking part in the micro tasks often receive very low wages, and 2) employment through these platforms puts the people to position, which would be considered unethical in first-world countries. Goodman et al. [16] on the other hand discuss the reliability of results obtained by crowd sourcing compared to traditional sampling. They conclude that despite differences between participants of crowd sourcing and traditional sampling, the results are reliable. However, they recommend 1) to include screening questions to ensure language comprehension and attention, 2) to avoid questions that require a factual answer, and 3) to keep in mind the possible effect on results caused by individual differences in financial and social domains. Both of the studies mentioned deal with MTurk, a crowd sourcing platform by Amazon, but generalize for other available platforms as well.

According to van der Lee et al. [46] there is basically no agreement on how NLG systems should be evaluated. Human evaluation is used often but in various different ways. This presents two major problems. First, results between different research groups are not comparable between each other. Second, it is difficult for those new to the field to know what the best practices are in terms of conducting the evaluation. They provide

an overview of the current state of human evaluation in NLG, and propose a set of best practices to follow.

Next, I summarize those best practices. Firstly, their general advice is that human evaluation should always be conducted, when possible. They advice that criteria used in evaluation should be properly defined, and that separate – task optimized – criteria, rather than an overall quality assessment, should be used. They for example note that the tasks of style transfer and generating weather reports might have different criteria. For the number of judges used, they advice to conduct a large-scale reader-focused evaluation rather than a small-scale expert focused one. Furthermore, they emphasise that the sample size and demographics should be reported and motivated. For annotation tasks, minimum of two judges, preferably more, should be used and the Inter-Annotator Agreement score should be reported. For a quantitative study, they advice to prefer a 7-point Likert scales or continuous ranking for measurement, as other studies discourage the use of smaller scales and have found that larger scales do not increase reliability. For tasks where a respondent evaluates multiple items, the items should be either randomly ordered or balanced to reduce order- and learning effects. Finally, when the evaluation study is exploratory, they advice to only report exploratory data analysis, while when the study is confirmatory, they advice to consider preregistering and conducting appropriate statistical analyses.

## 2.5 Context

For this thesis work, I concentrate on modifying news reports generated by a template-based modular NLG system. To place this work in a context, I will briefly describe the system itself. The system architecture of my EU NLG system is derived from template-based modular architecture presented by Leppänen et al. [23]. The system generates reports from time-series data provided by Eurostat – the statistical office of the European Union.

The system is able to present given data in several languages in a technically accurate manner with only a few templates. However, the language is very stiff, and the sentences are very alike, which makes the final report consisting of several sentences very repetitive.

The template-based approach was chosen for this system because 1) the data available is structured and no corresponding natural language text were available for training of machine learning models, and 2) the system is required to work for multiple languages, especially for low-resource ones.

Although this research context is narrow, I anticipate that the algorithms I propose would transfer well to other contexts as the approach expects full sentences as input. Thus, this approach could be useful with any NLG system.

# 3. Research design

As observed in Section 2.1, a significant downside of template-based NLG methods for news automation is the lack of variety in their output. I expect that a solution for this mentioned issue could be to increase the diversity of vocabulary used in the news report. For this thesis, I narrow down the investigation of this expectation to two scenarios. In the first scenario, a new word is embedded into a sentence to a pre-defined position with a pre-defined part-of-speech tag. In the second scenario, an existing word from the sentence is replaced with some other word. To meet the requirements of news media, I further narrow this down so that only those changes that retain the original meaning are considered successful. To achieve this, in the second scenario words are replaced with synonyms.

In this thesis, I propose an algorithm for each of the above scenarios. The algorithm that populates empty slots in sentences is referred to as lexicalization, and the one that replaces existing words is referred to as relexicalization. The algorithms are implemented for testing purposes to an existing template-based NLG system that is described earlier in Chapter 2. These algorithms are described in detail in Chapter 4.

Furthermore, the language resources used in my (re)lexicalization algorithms are available only for high-resource languages. I expect that systems operating on low-resource languages can benefit from my approaches to at least some extent via use of cross-lingual word embeddings introduced in Section 2.3.3.

To investigate the above expectation, I implement variations of the (re)lexicalization algorithms for low-resource languages. In these new algorithms, I translate low-resource language words to high-resource language words with cross-lingual word embeddings. These translations are used to utilise language resources available for the high-resource language. Otherwise these algorithms apply the (re)lexicalization approach similarly as for high-resourced languages. It is notable that my low-resource variants are dependent on availability of a BERT model. These algorithms are also described in detail in Chapter 4.

## 3.1 Research objectives

Based on the expectations described above, I form the following two research objectives for this thesis work.

1. To examine whether automatically adding new words and replacing existing ones could make automated news reports more pleasant for a human reader while retaining the original news content.

2. To examine whether linguistic resources of high-resource languages can be utilised for low-resource languages with help of cross-lingual word embeddings.

# 4. Algorithms

In this chapter I describe how the modifications to NLG outputs are done, and present the used algorithms as pseudocode.

Both of these modifications could be achieved by choosing words from user defined lists of approved words. This trivial solution however does not contribute to fulfilling the requirement of transferability of the system for automated journalism identified by Leppänen et al. [23], as a list constructed for one application domain (e.g. news) might not be applicable for another domain (e.g. technical manuals). Thus, for this thesis work I concentrate on developing approaches that would contribute to fulfilling that requirement as well. Furthermore, defining the (re)lexicalizations to templates is not notably more laborious than defining those static lists of approved words.

## 4.1 The word embedding models

This thesis work is dependent on word embedding models developed for purposes of EU funded EMBEDDIA research project. All of the algorithms described later in this chapter utilise the same trilingual BERT model, FinEst BERT [44]. In addition to the BERT model, the low-resource language variants of (re)lexicalization algorithms utilize cross-lingual word embedding mappings for word translations. The monolingual word embeddings are constructed with FastText [4] and mapped with VecMap [1] to form the cross-lingual embeddings.

The FinEst BERT model is trained with monolingual corpora for English, Finnish and Estonian. The corpora sizes are presented in table 4.1. The data is a mixture of news articles and general web crawl.

| **Language** | **Corpora size** |
|---|---:|
| English | 157 |
| Finnish | 97 |
| Estonian | 75 |

**Table 4.1:** The sizes of monolingual training corpora for the FinEst BERT model[44]. The sizes are expressed in millions of tokens used to create word piece vocabularies.

## 4.2 Lexicalization approach

In this section I describe the lexicalization approach, both high- and low-resource language variants. As described in Chapter 3, lexicalization is a method where new words are embedded to pre-defined positions in sentences, and possible words are filtered with pre-defined part-of-speech tags. The word is then taken by random from the final set. The slot is left empty when the proposed words do not fulfil the requirements or the set is empty.

### 4.2.1 High-resource language variant

The following pseudocode (Algorithm 1) describes the high-resource language variant of my lexicalization algorithm.

**Algorithm 1** Pseudocode describing the lexicalization approach sentences generated using template based NLG. The approach is tailored for high-resource languages, such as English, and uses additional linguistic resources (here, a part of speech tagger) to conduct further filtering.

**function** LEXICALIZEHIGHRESOURCEWITHPOSFILTER(
$Sentence$, $PosTag$, $k$, $minMasked$, $maxMasked$)
  $WordsAndScores \leftarrow \emptyset$
  **for** $n \in [minMasked, maxMasked]$ **do**
    $MaskedSentence \leftarrow$ Sentence with $n$ `[MASK]` tokens
    $Words, Scores \leftarrow MaskedLM.TopKPredictions(MaskedSentence, k)$
    $WordsAndScores \leftarrow WordAndScores \cup \{(w, s) | w \in Words$ **and** $s \in Scores\}$
  **end for**
  $Proposals \leftarrow \{w | (w, s) \in WordsAndScores$ **and** $PosTag(w) = PosTag$
    **and** $s >= Threshold\}$
  **return** SAMPLE($Proposals$)
**end function**

For this thesis work, the masked language model used is FinEst BERT model [44] described earlier in Section 4.1 and POS-tagging is done with the NLTK library [3].

BERT is used for predicting new suitable words for an empty slot in a sentence. This is achieved by placing one or multiple `[MASK]` tokens to the position of interest in the sentence. Multiple `[MASK]` tokens are required as words can be represented with subword tokens. When a prediction for masked tokens is made, BERT model returns score for each token in its vocabulary for each `[MASK]` token in the sentence. Thus, in the above algorithm, the masked language model gives word proposals and their scores.

These scores may be converted into likelihoods, scores that sum up to one, by using a softmax function.

### 4.2.2 Low-resource language variant

In this subsection I describe how the lexicalization approach described above is modified for a low-resource language where no POS-taggers are readily available. The algorithm is as the high-resource language variant (Algorithm 1), but non-contextual cross-lingual word embeddings are used for checking the part-of-speech tags. The used embeddings are described earlier in Section 4.1.

The low-resource words are vectorized and the most similar high-resource words are retrieved from the high-resource language vector space for each low-resource word. These high-resource language words are then POS-tagged and the resulting POS-tags are paired with the original low-resource words. See Algorithm 2 for how the POS-ragging is achieved for low-resource languages.

**Algorithm 2** Pseudocode describing how the language resources, here a POS-tagger, are utilized for a low-resource language with cross-lingual word embeddings. In other words, when working with a low-resource language, lexicalization is done as in Algorithm 1, but the POS-tagging phase utilises this algorithm.

**function** POSTAGLOWRESOURCELANGUAGE($LowResWord$, $LowResEmbeddings$, $HighResEmbeddings$)

$LowResVector \leftarrow$ FINDVECTOR($LowResWord, LowResEmbeddings$)

$HighResWord \leftarrow$ CLOSESTWORD($LowResVector, HighResEmbeddings$)

$LowResTagged \leftarrow (LowResWord,$ POSTAG($HighResWord$))

**return** $LowResTagged$

**end function**

## 4.3 Relexicalization approach

In this section I describe the relexicalization approach, where existing words in the sentence to be modified are replaced with new words. I restrict the search goal to synonyms to avoid predicting antonyms of the original word since replacing with an antonym would change the meaning tremendously. Antonyms are nearby vectors in relation to the original word vector in the vector space, as are synonyms.

I further restrict the changes to nouns, verbs, adjectives and adverbs since those are word classes available in WordNet [30]. WordNet is the lexical database in English that I

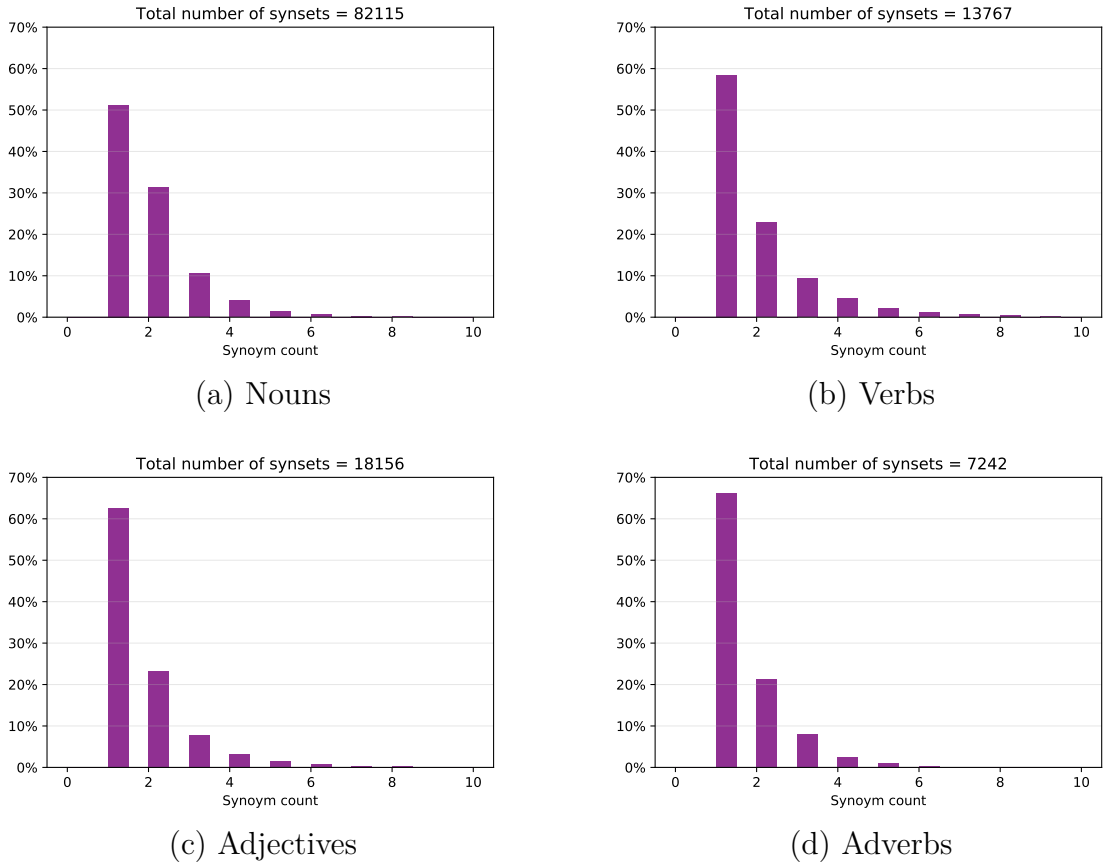


**Figure 4.1:** Histograms for number of synonyms per synset per word class in the lexical database for the English language, WordNet [30].

use for finding synonyms. Figure 4.1 presents the distributions of number of synonyms in synsets for each word class in WordNet. Synsets are sets of one or more synonyms. The y-axes are represented as percentages for easier comparison between the groups, as they differentiate in size. If a word belongs to a synset with size one, this means that it has no synonyms.

On the basis of these graphs, it can be assumed that replacing nouns would be most successful, because there are at least some synonyms available for around 50% of them. The relative number of synonyms slightly decreases for verbs, adjectives, and and finally for adverbs. Testing was done with nouns, verbs and adjectives, as the simple sentences used in testing did not contain adverbs.

During relexicalization a word might remain unchanged if there are no good options in the synonyms or by chance since the original word is also included in the set from which the replacement word is chosen. The original word is included as it maximises the number of options.

### 4.3.1 High-resource language variant

Algorithm 3 describes how synonyms are retrieved from a lexical database, here WordNet [30], and how they are scored with a masked language model for a sentence. The masked language model used is FinEst BERT [44] described earlier in Section 4.1.

**Algorithm 3** Pseudocode describing a method for relexicalizing using a combination of a masked language model (based on contextual word embeddings) and a synonym dictionary, such as provided by WordNet.

**function** RELEXICALIZEHIGHRESOURCEWITHSYNONYMS($OriginalWord, Sentence$)

$WordsAndScores \leftarrow \emptyset$

$Synonyms \leftarrow$ GETSYNONYMS($OriginalWord$)

**for** $w \in Synonyms$ **do**

$CandidateSentence \leftarrow$ Sentence with $w$ replacing the original word

$CandidateScore \leftarrow MaskedLM.Score(CandidateSentence, w)$

$WordsAndScores \leftarrow WordsAndScores \cup (w, CandidateScore)$

**end for**

$Proposals \leftarrow \{w|(w, s) \in WordsAndScores$ **and** $s >= Threshold\}$

**return** SAMPLE($Proposals$)

**end function**

Retrieving synsets for a word provides all synonyms with different semantics. I rely on BERT to filter out the possible unfitting options. A subset of the list of synonyms is taken by filtering with a threshold value for the likelihood of each option to appear in the specific context - the sentence. The final word is chosen from this subset randomly, as this would result in the most varied language, assuming that the words in the list are all sufficiently fitting.

When the candidate word is scored with BERT, it is tokenized. A word that is not part of BERT's fixed size vocabulary is tokenized with multiple subword tokens. An example of this, in the BERT model I used, is the word 'absolutely'. It is tokenized to two tokens: 'absolute' and '##ly'. BERT returns individual score for each subword token. In this thesis, a mean is taken from those scores to get the word score.

### 4.3.2 Low-resource language variant

In this subsection I describe how I harnessed cross-lingual word embeddings to find synonyms for a low-resource language where there is no synonym dictionary readily available.

The low-resource language variant of relexicalization algorithm is as in Algorithm 3,

but a round-trip via a high-resource language is done when retrieving synonyms. The low-resource language words are translated to high-resource language, after which synonyms for these translations are retrieved from the synonym dictionary available in the high-resource language. The synonyms are then translated back to the low-resource language.

**Algorithm 4** Pseudocode describing how synonyms are retrieved for a low-resource language by utilizing cross-lingual word embeddings. Low-resource variant of relexicalization is as Algorithm 3, but this algorithm is used to retrieve synonyms.

```
function SynonymsForLowResourceLanguage(
LowResWord, LowResEmbeddings, HighResEmbeddings)
    LowResVector ← FindVector(LowResWord, LowResEmbeddings)
    HighResWord ← ClosestWord(LowResVector, HighResEmbeddings)
    HighResSynonyms ← GetSynonyms(HighResWord)
    LowResSynonyms ← ∅
    for w ∈ HighResSynonyms do
        HighResVector ← FindVector(w, HighResEmbeddings)
        LowResWord ← ClosestWord(HighResVector, LowResEmbeddings)
        LowResSynonyms ← LowResSynonyms ∪ {LowResWord}
    end for
    return LowResSynonyms
end function
```

The closest word from aligned cross-lingual word embeddings seems to often be a direct translation of the original word. Retrieving more than one of the closest words would lead to a wider set of candidate words. Thus, I conducted testing with the three ways listed below.

1. Retrieve $N$ most similar high-resource language words (where $N > 1$). Then retrieve synonyms from WordNet for the words. Finally retrieve $N$ most similar low-resource language words for the set synonyms (where $N > 1$).

2. Retrieve $N$ most similar high-resource language words (where $N = 1$). Then retrieve synonyms from WordNet for the words. Finally retrieve $M$ most similar low-resource language words for the set synonyms (where $M > 1$).

3. Retrieve $N$ most similar high-resource language words (where $N = 1$). Then retrieve synonyms from WordNet for the words. Finally retrieve $N$ most similar low-resource language words for the set synonyms (where $N = 1$).

,

A problem that arises with options 1 and 2, is that some of the closest words given by cross-lingual aligned word embedding are the antonyms we wish to avoid. Finally, I ended up using option number 3.

During the work I noticed that for morphologically rich languages such as Finnish, my algorithm leads to better synonym sets when retrieving them for words in their lemmatized form. This is due to the synonyms being in their lemmatized, or "dictionary", form in the synonym dictionary. As a result of this finding, I apply UralicNLP API [18] to lemmatize the original word, and to analyze the original morphology and generate it back to the retrieved synonyms. Oftentimes, a word can be lemmatized in multiple ways. In my approach, synonyms are retrieved for all of the lemmas. Then, the algorithm tries to regenerate all morphologies proposed by UralicApi for all synonyms. I present example analyzes in Figure 4.2. Some of the results are ungrammatical or contextually incorrect but I rely on BERT to score these unlikely.

a) vanha+A+Comp+Pl+Gen

b) vanhempi+A+Pl+Gen

c) vanhempi+N+Pl+Gen

**Figure 4.2:** Possible morphologies of the Finnish word ‘vanhempien’. This word has two lemmas: ‘vanha’ (Eng. old) and ‘vanhempi’ (Eng. parent or older). The correct analysis depends on the context.

# 5. Empirical results and discussion

In this chapter I will present the empirical results of my research and discuss them. The evaluation is designed to reflect the research objectives laid out in chapter 3: 1) to examine whether automatically adding new words and replacing existing ones could make automated news reports more pleasant for a human reader while retaining the original news content, and 2) to examine whether linguistic resources of high-resource languages can be utilised for low-resource languages with help of cross-lingual word embeddings.

Therefore, with this evaluation, I aim to investigate whether my algorithms produce fluent sentences and retain the original news content essentially. The investigation is done with both high- and low-resource languages.

## 5.1 Evaluation setup

In this section I describe how the obtained results are evaluated. I follow the best practices for human evaluation of NLG systems described by van der Lee et al. [46]. These best practices are summarised in Section 2.4.

For evaluation, human judges are presented evaluation groups that consist of three statements and two questions. The three statements are about a pair of sentences. Sentence 1 is the original sentence with either an empty slot in it that needs to be lexicalized or one word marked to be relexicalized. Sentence 2 is the (re)lexicalized sentence. The judge is asked to evaluate the following statements on a 7-point Likert scale (Table 5.1):

1. Sentence 1 is a good quality sentence in the target language.
2. Sentence 2 is a good quality sentence in the target language.
3. Sentences 1 and 2 have essentially the same meaning.

Statement 1 and 2 are presented to evaluate if the modification has an effect to the output quality. Statement 3 is presented to evaluate if meaning was preserved as it was intended to. The 7-point Likert scale is chosen based on recommendation of van der Lee et al. [46] as they state that based on experimental literature it maximises reliability, validity and discriminative power of results.

| Score | Label |
|---|---|
| 1 | Strongly disagree |
| 2 | Quite strongly disagree |
| 3 | Somewhat disagree |
| 4 | Neither agree nor disagree |
| 5 | Somewhat agree |
| 6 | Quite strongly agree |
| 7 | Strongly agree |

**Table 5.1:** 7-point Likert scale used for statements 1, 2 and 3.

In addition to those three statements, the respondent is asked two questions, both of which concern a separate group of words. Word group 1 contains the words from which the used word was chosen from. Word group 2 contains the words which were ruled out by the system. To further clarify, all words in both groups met the criteria of being synonyms or being the correct part-of-speech. The division to accepted and unaccepted words was done based on the score given by BERT. The aim here was to examine if using the score by BERT would rule out unfitting words. The judge is asked to evaluate the following questions on a 5-point Likert scale (Table 5.2):

4. How many of the words in word group 1 could be used in the marked place in sentence 1 so that the meaning remains essentially the same?

5. How many of the words in word group 2 could be used in the marked place in sentence 1 so that the meaning remains essentially the same?

The 5-point Likert scale was chosen while two test judges took the quiz in Finnish and found 7-point scale hard to understand for these questions. Sizes of the word groups described above are limited to six items for quicker and less exhausting answering.

| Score | Label |
|---|---|
| 1 | None of the words |
| 2 | Less than half of the words |
| 3 | Half of the words |
| 4 | More than half of the words |
| 5 | All of the words |

**Table 5.2:** 5-point Likert scale used for questions 4 and 5.

For expert evaluation just a few judges is enough but for reader-focused style evaluation less than 50 is considered bad, 100 or more is good [46]. For English results, I was

able to gather 300 judgements for 100 evaluation groups described above. This means that three judgements per an evaluation group were given. This was possible by using a crowd-sourcing platform called Appen, where the judges received a monetary reward. For Finnish results, the evaluation setup was less optimal due to lack of a usable platform: 20 evaluation groups each got 21 judgements when the judges all gave a judgement per a group. The Finnish judges were recruited via a student mailing list and via my own network.

Whatever the sample size, according to van der Lee et al, a minimum good practice guideline is to always report participant numbers, with relevant demographic data (i.e., gender, nationality, age, fluency in the target language, academic background, etc). I present the demographics for my sample of Finnish results in tables 5.3, 5.4 and 5.5. Unfortunately the platform in which English results were gathered does not enable these kind of questions. All of the English respondents were from the USA.

| **Educational background** | **Count** | **Percentage** |
|---|---|---|
| Basic level education | 0 | 0% |
| Secondary education | 5 | 24% |
| Bachelor's degree | 11 | 52% |
| Master's degree | 5 | 24% |
| **Total** | **21** | **100%** |

**Table 5.3:** Highest level of education completed by the Finnish judges.

| **Age group** | **Count** | **Percentage** |
|---|---|---|
| 18-25 | 5 | 24% |
| 26-35 | 11 | 52% |
| 36-45 | 3 | 14% |
| 46-55 | 2 | 10% |
| 56 or older | 0 | 0% |
| **Total** | **21** | **100%** |

**Table 5.4:** Age groups of the Finnish judges.

| **Fluency in Finnish** | **Count** | **Percentage** |
|---|---|---|
| Native speakers | 20 | 95% |
| Working proficiency | 1 | 5% |
| Lower language skills | 0 | 0% |
| **Total** | **21** | **100%** |

**Table 5.5:** Level of fluency in Finnish of the Finnish judges.

For evaluation, I ruled out cases where the sentence would remain unchanged. Due to the design of my evaluation setup, I also ruled out scenarios, where either the remaining set of approved words after word selection or the set of disapproved word were empty. The two test judges found these scenarios difficult to comprehend, as the survey would ask them to evaluate if there are fitting words for a sentence in an empty set (see questions 4 and 5 described above).

| **POS tag** | **Tag name** |
|---|---|
| RB | Adverb |
| IN | Preposition |
| NN | Noun |
| WRB | Wh-adverb |
| DT | Determiner |
| WP | Wh-pronoun |
| JJ | Ajective |

**Table 5.6:** The POS tags present in evaluations.

## 5.2 High-resource (re)lexicalization approach

In this section I discuss the results of the high-resource language variant of my (re)lexicalization approach and aim to reflect my first research objective of whether automatically adding new words and replacing existing ones could make automated news reports more pleasant for a human reader while retaining the original news content.

a) In May 2018, however the growth rate on previous month was for the category housing, water, electricity, and gas and other fuels 0.6.

b) In Austria in 2018 75 year old or older females still received median equivalised net income of 22234 €.

c) In 2017, 65 year old or older females did ~~earn~~ bring in mean net income of 27871 €.

d) In Finland in 2016 households' total ~~expenditure~~ spending on healthcare was 20.35 %.

**Figure 5.1:** Example sentences a and b are produced using the lexicalization method in English. The underlined token was added during lexicalization. The sentences were scored high quality by the judges. Example sentences c and d are produced using the relexicalization method in English. The section modified by relexicalization is denoted by underlining, with the original seed word struck over and the replacement word shown without strikethrough. The sentences were scored high quality by the judges.

Figure 5.2 shows the aggregated results for the high-resource language (English) variant of the lexicalization approach. The results indicate that the sentence meanings remained essentially the same after the modifications. In addition, there are very minor differences observable between the results for different POS-tags (Table 5.7).

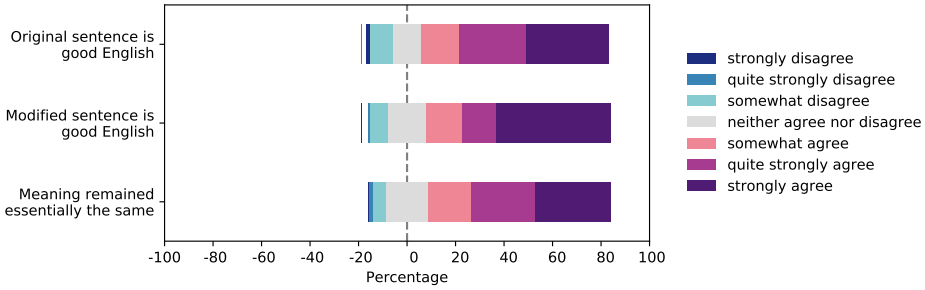


**Figure 5.2:** Quality of sentences with high-resource lexicalization in English, and preservation of sentence meaning. Here, the results for all part-of-speech tags are aggregated together.

Similarly, Figure 5.3 shows the aggregated results for high-resource variant of the relexicalization approach. The results for this approach as well show that the quality was preserved, yet not improved as above. This might be a due to respondents finding sentences with more words better. The quality remaining the same is a success, as it indicates that the method was able to create variety without compromising the quality. It is similarly positive that the meaning was considered to remain essentially the same. In addition, similarly as above, there are very minor differences observable between these results for different word classes (Table 5.8).

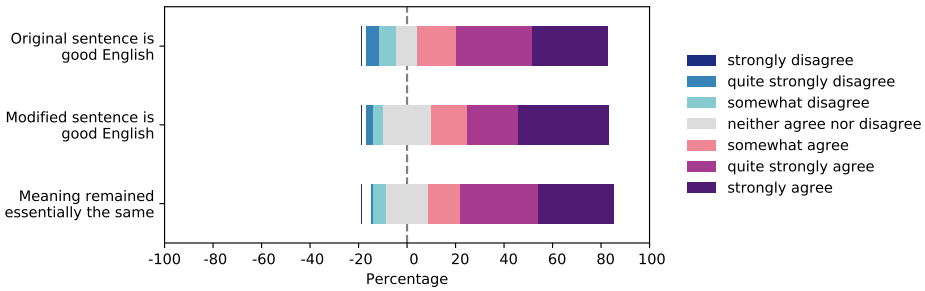


**Figure 5.3:** Quality of sentences with high-resource relexicalization in English, and preservation of sentence meaning. Here, the results for all word classes are aggregated together.

| POS | R | J/R | Statement | Mean | Std |
|---|---|---|---|---|---|
| All | 50 | 3 | Original sentence is good English | 5.57 | 1.46 |
| | | | Modified sentence is good English | 5.76 | 1.41 |
| | | | Meaning remained essentially the same | 5.54 | 1.36 |
| RB | 10 | 3 | Original sentence is good English | 5.67 | 1.47 |
| | | | Modified sentence is good English | **5.93** | 1.46 |
| | | | Meaning remained essentially the same | 5.53 | 1.45 |
| IN | 9 | 3 | Original sentence is good English | 5.78 | 1.12 |
| | | | Modified sentence is good English | 5.89 | 1.13 |
| | | | Meaning remained essentially the same | 5.44 | 1.42 |
| NN | 9 | 3 | Original sentence is good English | 5.70 | 1.49 |
| | | | Modified sentence is good English | 5.78 | 1.39 |
| | | | Meaning remained essentially the same | 5.70 | 1.06 |
| WRB | 4 | 3 | Original sentence is good English | 5.08 | 1.62 |
| | | | Modified sentence is good English | 5.42 | 1.62 |
| | | | Meaning remained essentially the same | 5.25 | 1.29 |
| DT | 8 | 3 | Original sentence is good English | 5.54 | 1.59 |
| | | | Modified sentence is good English | 5.92 | 1.38 |
| | | | Meaning remained essentially the same | **5.79** | 1.50 |
| WP | 1 | 3 | Original sentence is good English | 6.00 | 1.00 |
| | | | Modified sentence is good English | 5.67 | 1.53 |
| | | | Meaning remained essentially the same | 5.33 | 0.58 |

**Table 5.7:** Quality of sentences with high-resource lexicalization in English, and preservation of sentence meaning. The Likert scale for each statement is 7-point. R = number of statement triplets evaluated, J/R = judgements per row. Best scores are highlighted with bold. The topmost row corresponds to Figure 5.2. The POS tags are explained in Table 5.6.

| WC | R | J/R | Statement | Mean | Std |
|---|---|---|---|---|---|
| All | 50 | 3 | Original sentence is good English | 5.55 | 1.46 |
| | | | Modified sentence is good English | 5.60 | 1.40 |
| | | | Meaning remained essentially the same | 5.65 | 1.27 |
| Noun | 23 | 3 | Original sentence is good English | 5.71 | 1.36 |
| | | | Modified sentence is good English | **5.78** | 1.40 |
| | | | Meaning remained essentially the same | **5.71** | 1.23 |
| Verb | 14 | 3 | Original sentence is good English | 5.50 | 1.49 |
| | | | Modified sentence is good English | 5.62 | 1.28 |
| | | | Meaning remained essentially the same | 5.67 | 1.26 |
| Adj. | 14 | 3 | Original sentence is good English | 5.31 | 1.61 |
| | | | Modified sentence is good English | 5.26 | 1.52 |
| | | | Meaning remained essentially the same | 5.51 | 1.37 |

**Table 5.8:** Quality of sentences with high-resource relexicalization in English, and preservation of sentence meaning. The Likert scale for each statement is 7-point. R = number of statement triplets evaluated, J/R = judgements per row. Best scores are highlighted with bold. The topmost row corresponds to Figure 5.3. WC stands for word class.

With both approaches, results for questions about the word groups show that many words in the approved set were unfitting and similarly many words in the disapproved set were fitting (Figure 5.4 for lexicalization and Figure 5.5 for relexicalization). As mentioned in Section 5.1, all words in both groups met the criteria of being synonyms (relexicalization) or being the correct part-of-speech (lexicalization), but the division to accepted and unaccepted words was done based on the likelihood score given by BERT. This suggests, that the current filtering method, where these likelihoods are expected to indicate, whether a word is fitting for a context, is unsuccessful. The results are described in detail in Table 5.9 and Table 5.14.

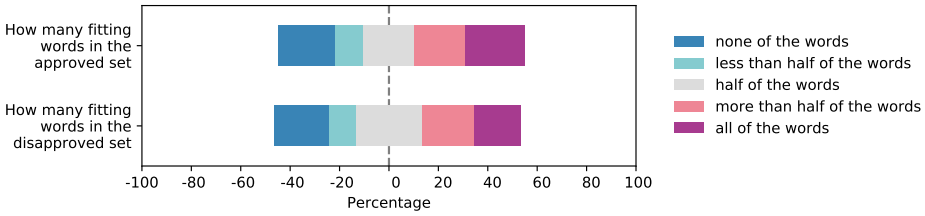


**Figure 5.4:** Quality of groups of approved and disapproved words with high-resource lexicalization in English. Here, the results for all part-of-speech tags are aggregated together.

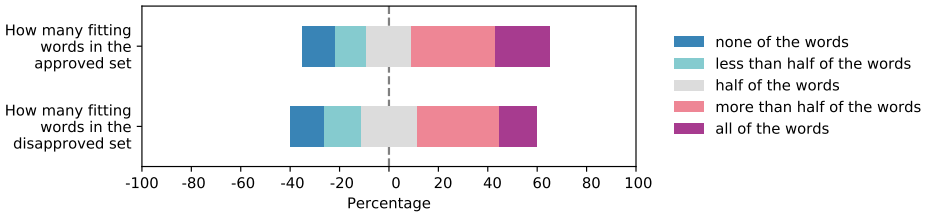


**Figure 5.5:** Quality of groups of approved and disapproved words with high-resource relexicalization in English. Here, the results for all word classes are aggregated together.

Based on the above observations, I dare to say that the high-resource (re)lexicalization approach is able to produce varied output in a way consistent with the requirements of automated journalism. To further clarify, the output remains accurate and transparent, and the system remains transferable and modifiable. I further anticipate that the results would improve and be more reliable, if the filtering method were to be improved.

A notable matter is that the results for relexicalization approach only reflect cases where there were multiple proposed words. In other words, all word groups presented in evaluation, had at least one word other than the original left after choosing a word for the sentence. However, when there are only few synonyms proposed, the end result would most likely be better, as the synonyms would more likely have the same sense. To clarify, as all synsets for a word are retrieved and considered as synonyms in my approach, it is highly probable that if the number of synonyms is high, some of them have a wrong sense.

| POS | R | J/R | Statement | Mean | Std |
|---|---|---|---|---|---|
| All | 50 | 3 | How many fitting words in the approved set | 3.11 | 1.49 |
| | | | How many fitting words in the disapproved set | 3.03 | 1.41 |
| RB | 10 | 3 | How many fitting words in the approved set | 3.30 | 1.37 |
| | | | How many fitting words in the disapproved set | 3.23 | 1.35 |
| IN | 9 | 3 | How many fitting words in the approved set | 2.78 | 1.31 |
| | | | How many fitting words in the disapproved set | 2.92 | 1.27 |
| NN | 9 | 3 | How many fitting words in the approved set | 3.07 | 1.66 |
| | | | How many fitting words in the disapproved set | 3.15 | 1.46 |
| WRB | 4 | 3 | How many fitting words in the approved set | 3.17 | 1.64 |
| | | | How many fitting words in the disapproved set | 3.42 | 1.56 |
| DT | 8 | 3 | How many fitting words in the approved set | 3.29 | 1.55 |
| | | | How many fitting words in the disapproved set | 2.67 | 1.52 |
| WP | 1 | 3 | How many fitting words in the approved set | 4.67 | 0.58 |
| | | | How many fitting words in the disapproved set | 4.33 | 0.58 |

**Table 5.9:** Quality of groups of approved and disapproved words with high-resource lexicalization in English. The Likert scale for each question is 5-point. R = number of data rows evaluated, J/R = judgements per row. The topmost row corresponds to Figure 5.4.

| WC | R | J/R | Statement | Mean | Std |
|---|---|---|---|---|---|
| All | 50 | 3 | How many fitting words in the approved set | 3.39 | 1.32 |
| | | | How many fitting words in the disapproved set | 3.21 | 1.27 |
| Noun | 23 | 3 | How many fitting words in the approved set | 3.43 | 1.24 |
| | | | How many fitting words in the disapproved set | 3.20 | 1.35 |
| Verb | 14 | 3 | How many fitting words in the approved set | 3.40 | 1.33 |
| | | | How many fitting words in the disapproved set | 2.88 | 1.40 |
| Adj. | 14 | 3 | How many fitting words in the approved set | 3.28 | 1.47 |
| | | | How many fitting words in the disapproved set | 3.59 | 0.85 |

**Table 5.10:** Quality of groups of approved and disapproved words with high-resource relexicalization in English. The Likert scale for each question is 5-point. R = number of data rows evaluated, J/R = judgements per row. WC stands for word class. The topmost row corresponds to Figure 5.5.

## 5.3 Low-resource (re)lexicalization approach

In this section I present and discuss the results of the low-resource language variant of my (re)lexicalization approach, where a round-trip to a high-resource language is conducted using cross-lingual word embeddings. With this discussion I aim to reflect on my second research objective of whether linguistic resources of high-resource languages can be utilised for low-resource languages with help of cross-lingual word embeddings.

a) Toisaalta Ruotsissa vuonna 2018 55-64-vuotiaiden naisten tulojen mediaani oli 313792 paikallisessa valuutassa ilmaistuna.

b) Maltalla vuonna 2015 kotitaloudet maksoivat omaa terveydenhuollostaan itse 37.47 %.

c) Kyproksella vuonna 2017 ~~kotitaloudet~~ perheet maksoivat terveydenhuollon menoistaan itse 44.64 %.

d) Ranskassa vuonna 2017 75-vuotiaiden ja ~~vanhempien~~ vanhuksien naisten tulojen keskiarvo oli 25770 €.

**Figure 5.6:** Example sentences (a) and (b) are produced using the lexicalization method in Finnish. The underlined tokens were added during lexicalization. Sentence (a) was scored high by the judges and it translates as '*On the other hand, in Sweden in 2018 the median income of females between ages 55 and 64 was 313792 when expressed in national currency.*'. Sentence (b) was scored low as it contains an ungrammatical token. It translates roughly as '*In Malta in 2015, households paid out-of-pocket 37.47 % of their selves own healthcare.*' Example sentences (c) and (d) are produced using the relexicalization method in Finnish. The section modified by relexicalization is denoted by underlining, with the original seed word struck over and the replacement word shown without strikethrough. Sentence (c) translates as '*In Cyprus in 2017, ~~households~~ families paid 44.64 % of their health care expenses themselves.*', and was scored well by the judges. Sentence (d) was scored low as it contains an ungrammatical token. It translates roughly as '*In France in 2017 the mean income of females aged 75 or ~~older~~ elderly was 25770 €.*'

Figure 5.7 shows the aggregated results for the low-resource variant of the lexicalization approach. These results indicate that the modifications did compromise the quality of the sentences. A major issue with Finnish compared to English is the complex morphology of Finnish language – BERT has trouble coming up with words that would be in a contextually correct morphological form. It is also notable that, according to the judges, the sentence meaning changed significantly in some cases.

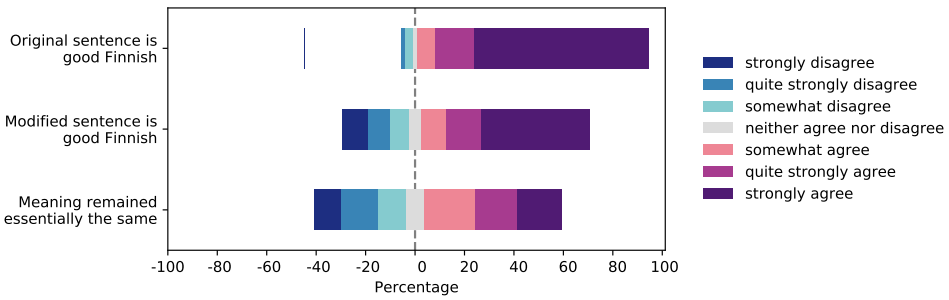


**Figure 5.7:** Quality of sentences with low-resource lexicalization in Finnish with English as the high-resource language, and preservation of sentence meaning. Here, the results for all part-of-speech tags are aggregated together.

| POS | R | J/R | Statement | Mean | Std |
|---|---|---|---|---|---|
| All | 10 | 21 | Original sentence is good Finnish | 6.43 | 0.88 |
| | | | Modified sentence is good Finnish | 5.12 | 1.36 |
| | | | Meaning remained essentially the same | 4.34 | 1.61 |
| DT | 2 | 21 | Original sentence is good Finnish | 6.45 | 0.98 |
| | | | Modified sentence is good Finnish | 6.14 | 1.24 |
| | | | Meaning remained essentially the same | 3.67 | 1.54 |
| IN | 2 | 21 | Original sentence is good Finnish | 5.93 | 1.42 |
| | | | Modified sentence is good Finnish | 5.38 | 1.66 |
| | | | Meaning remained essentially the same | 4.40 | 1.50 |
| JJ | 2 | 21 | Original sentence is good Finnish | 6.86 | 0.34 |
| | | | Modified sentence is good Finnish | 3.00 | 1.55 |
| | | | Meaning remained essentially the same | 4.76 | 2.07 |
| NN | 2 | 21 | Original sentence is good Finnish | 6.12 | 1.19 |
| | | | Modified sentence is good Finnish | 4.48 | 1.60 |
| | | | Meaning remained essentially the same | 3.95 | 1.50 |
| RB | 2 | 21 | Original sentence is good Finnish | 6.81 | 0.49 |
| | | | Modified sentence is good Finnish | **6.62** | 0.74 |
| | | | Meaning remained essentially the same | **4.95** | 1.45 |

**Table 5.11:** Quality of sentences with low-resource lexicalization in Finnish with English as the high-resource language, and preservation of sentence meaning. The Likert scale for each statement is 7-point. R = number of data rows evaluated, J/R = judgements per row. Best scores are highlighted with bold. The topmost row corresponds to Figure 5.7 and the row where POS = RB to Figure 5.8.

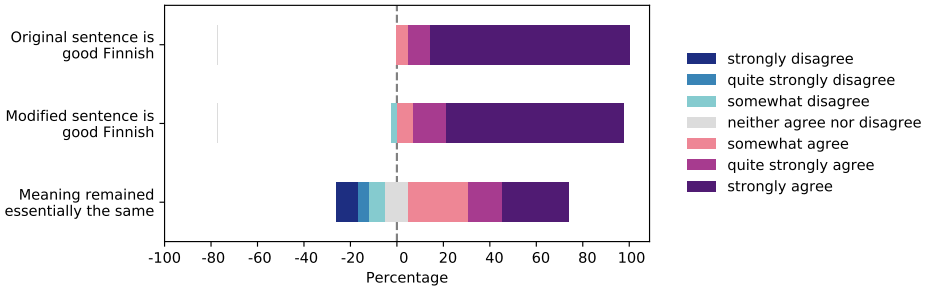


**Figure 5.8:** Quality of sentences with low-resource lexicalization in Finnish with English as the high-resource language, and preservation of sentence meaning. Here POS = RB (adverb).

Table 5.11 presents the detailed results for low-resource variant lexicalization. It shows that there is more variance in the results between POS-tags, than there were with the high-resource variant of the lexicalization approach. Figure 5.8 represents the results of the *best* POS-tag, adverb. These results are notably better than the aggregated results discussed above.

Figure 5.9 shows the results for the low-resource relexicalization variant. The same results are described in more detail in Table 5.12. Similarly, as for the lexicalization approach, the results show that the modifications did compromise the quality of the sentences. A change in meaning was also observed.

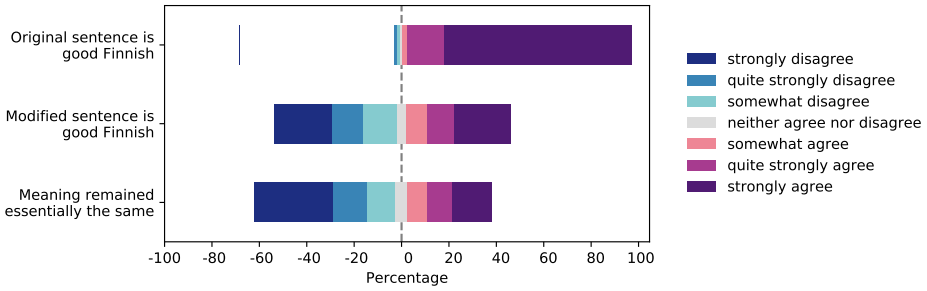


**Figure 5.9:** Quality of sentences with low-resource relexicalization in Finnish with English as the high-resource language, and preservation of sentence meaning. Here, the results for all part-of-speech tags are aggregated together.

| WC | R | J/R | Statement | Mean | Std |
|---|---|---|---|---|---|
| All | 10 | 21 | Original sentence is good Finnish | 6.67 | 0.66 |
| | | | Modified sentence is good Finnish | 3.89 | 1.43 |
| | | | Meaning remained essentially the same | 3.39 | 1.30 |
| Noun | 3 | 21 | Original sentence is good Finnish | 6.83 | 0.39 |
| | | | Modified sentence is good Finnish | **4.75** | 1.59 |
| | | | Meaning remained essentially the same | 3.04 | 1.11 |
| Verb | 4 | 21 | Original sentence is good Finnish | 6.68 | 0.67 |
| | | | Modified sentence is good Finnish | 4.14 | 1.41 |
| | | | Meaning remained essentially the same | **3.47** | 1.58 |
| Adj. | 3 | 21 | Original sentence is good Finnish | 6.51 | 0.92 |
| | | | Modified sentence is good Finnish | 2.70 | 1.32 |
| | | | Meaning remained essentially the same | 2.27 | 1.14 |

**Table 5.12:** Quality of sentences with low-resource relexicalization in Finnish with English as the high-resource language, and preservation of sentence meaning. Here, the word class is noun. The Likert scale for each statement is 7-point. R = number of data rows evaluated, J/R = judgements per row. Best scores are highlighted with bold. WC stands for word class. The topmost row corresponds to Figure 5.9.

A major reason for the low quality of the modified sentences was the attempt to bring back the correct morphological form with the lemmatization approach described in Section 4.3.2. The results of this method were often unsuccessful – the regeneration of morphology led both to technically correct but contextually incorrect morphological forms, and to ungrammatical words. Furthermore, BERT did not manage to score the likelihood of these contextually incorrect or ungrammatical morphological forms low, i.e. BERT considered the unfitting words fitting. Thus, by improving the restoring of morphology, I could improve the results for this low-resource lexicalization approach, as there would be less ungrammatical words for BERT to pick.

Figure 5.10 and Figure 5.11 shows the results for single word classes where best results on sentence quality and preservation of meaning were observed for low-resource-language in Finnish with English as the high-resource language.

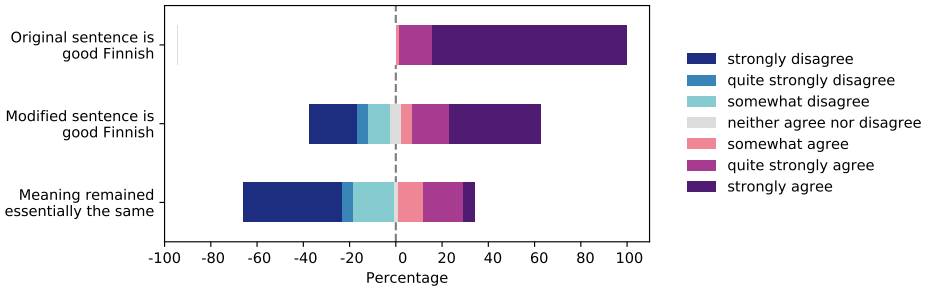


**Figure 5.10:** Quality of sentences with low-resource relexicalization in Finnish with English as the high-resource language, and preservation of sentence meaning. Here, the word class is noun.

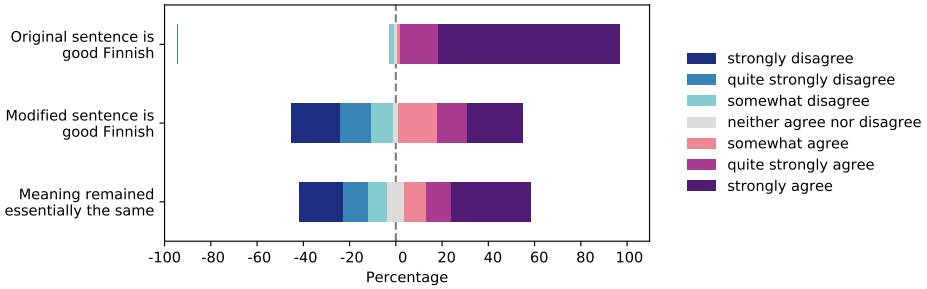


**Figure 5.11:** Quality of sentences with low-resource relexicalization in Finnish with English as the high-resource language, and preservation of sentence meaning. Here, the word class is verb.

The results on quality of the approved and disapproved groups of words with low-resource relexicalization in Finnish with English as the high-resource language are presented in Figure 5.12 and Figure 5.13. The figures show that there were very little fitting words available to begin with. I assume that this is again due to issues with Finnish morphology. These results are described in more detail in Table 5.13 and Table 5.14.

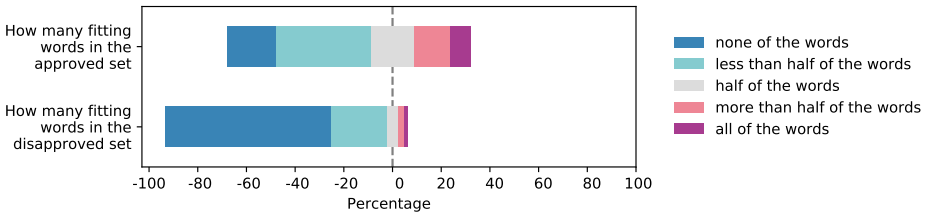


**Figure 5.12:** Quality of groups of approved and disapproved words with low-resource lexicalization in Finnish with English as the high-resource language. Here, the results for all word classes are aggregated.

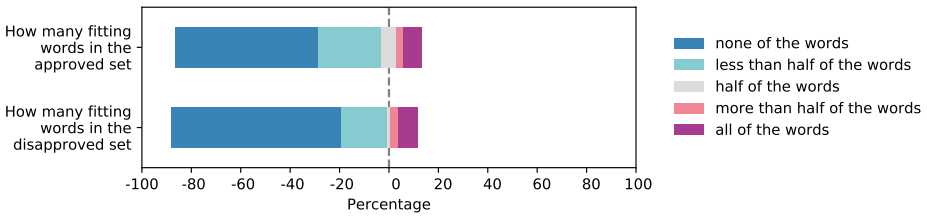


**Figure 5.13:** Quality of groups of approved and disapproved words with low-resource relexicalization in Finnish with English as the high-resource language. Here, the results for all word classes are aggregated.

| POS | R | J/R | Statement | Mean | Std |
|---|---|---|---|---|---|
| All | 10 | 21 | How many fitting words in the approved set | 2.53 | 0.82 |
| | | | How many fitting words in the disapproved set | 1.46 | 0.62 |
| DT | 2 | 21 | How many fitting words in the approved set | 3.52 | 0.81 |
| | | | How many fitting words in the disapproved set | 1.69 | 0.71 |
| IN | 2 | 21 | How many fitting words in the approved set | 1.55 | 0.76 |
| | | | How many fitting words in the disapproved set | 1.21 | 0.58 |
| JJ | 2 | 21 | How many fitting words in the approved set | 2.36 | 0.87 |
| | | | How many fitting words in the disapproved set | 1.10 | 0.28 |
| NN | 2 | 21 | How many fitting words in the approved set | 2.12 | 0.73 |
| | | | How many fitting words in the disapproved set | 1.55 | 0.63 |
| RB | 2 | 21 | How many fitting words in the approved set | 3.10 | 0.92 |
| | | | How many fitting words in the disapproved set | 1.74 | 0.89 |

**Table 5.13:** Quality of groups of approved and disapproved words with low-resource lexicalization in Finnish with English as the high-resource language. The Likert scale for each question is 5-point. R = number of data rows evaluated, J/R = judgements per row. WC stands for word class. The topmost row corresponds to Figure 5.12.

| WC | R | J/R | Question | Mean | Std |
|---|---|---|---|---|---|
| All | 10 | 21 | How many fitting words in the approved set | 1.76 | 0.78 |
| | | | How many fitting words in the disapproved set | 1.62 | 0.76 |
| Noun | 3 | 21 | How many fitting words in the approved set | 2.11 | 1.00 |
| | | | How many fitting words in the disapproved set | 1.49 | 0.67 |
| Verb | 4 | 21 | How many fitting words in the approved set | 1.57 | 0.69 |
| | | | How many fitting words in the disapproved set | 1.96 | 0.82 |
| Adj. | 3 | 21 | How many fitting words in the approved set | 1.67 | 0.68 |
| | | | How many fitting words in the disapproved set | 1.30 | 0.76 |

**Table 5.14:** Quality of groups of approved and disapproved words with low-resource relexicalization in Finnish with English as the high-resource language. The Likert scale for each question is 5-point. R = number of data rows evaluated, J/R = judgements per row. WC stands for word class. The topmost row corresponds to Figure 5.13.

Based on these observations, I interpret that the low-resource (re)lexicalization approach is currently unable to produce varied quality output reliably. However, as the results are notably better with some POS-tags, and as there is clear development area with the regeneration of morphology, I remain positive about the potential of this approach to fulfill my second research objective of whether linguistic resources of high-resource languages can be utilised for low-resource languages with help of cross-lingual word embeddings.

An observation I made during my thesis work is that some characteristics of Finnish make it possibly a relatively difficult language to handle with my (re)lexicalization methods. Firstly, it is very common to use compound words in Finnish, which is not the case in English. The methods led to better results when I avoided the use of compound words in Finnish templates. Secondly, as mentioned above, the complex morphology of Finnish demands additional handling and thus makes the method more prone to errors.

A notable matter with use of non-contextual cross-lingual word embeddings for this low-resource (re)lexicalization approach is that due to issues of non-contextual word embedding mentioned in Section 2.3, the translation achieved by taking the closest word from target language is not always correct. Then of course the set of synonyms is wrong as well.

# 6. Conclusions

In this thesis I have proposed two approaches for increasing variety of language in NLG system output in context of news – lexicalization and relexicalization. Both of the approaches utilise contextual word embeddings. In addition, I have proposed variants of (re)lexicalization which utilise cross-lingual word embeddings to benefit from linguistic resources of high-resource languages when working with low-resource languages.

The human evaluation conducted suggests that the high-resource variants of my (re)lexicalization approaches are already promising, as using them did create variety, but preserved quality and meaning. However, the results could be further improved by refining the filtering methods. While the human evaluations for low-resource variants of (re)lexicalization approaches did not show as good results, as decrease in quality and preservation of meaning were observed, there still are some glimpses of promising results and a clear development area on regeneration of morphology. This allows me to remain positive about the potential of my approaches.

## 6.1 Limitations of the study

The research and evaluation design appear to support the research objectives set for this initial study in the (re)lexicalization approach. Simultaneously, it is possible that some of the respondents have misunderstood some of the prompts in the evaluation questionnaire. However, as there were multiple judgements, the results should be reliable regardless of a few problem cases.

I anticipate that the results should generalize well to other domains, as no task specific optimization was done. However, this should be investigated for confirmation, as the domain of this study is very limited with a narrow domain of reports about EU statistics.

Finally, as I only evaluated the approaches with one high-resource language, English, and one low-resource language, Finnish, more evaluation should be conducted to validate behaviour with other languages. Issues with the complex morphology of Finnish hint that a low-resource language with less complex morphology (e.g. Yoruba with over 30 million speakers, most of them in Nigeria) might obtain better results. Furthermore, I

anticipate that some level of morphological analysis and regeneration should be applied for high-resource languages as well. This would promote the appropriate behaviour with a high-resource language with more complex morphology than English. English language allowed designing the templates in a way that avoided problems with morphology.

## 6.2 Future work

As the scope for this thesis is limited, interesting low hanging fruits remain for future research. In this section I lay out ideas that have come up during my work.

Firstly, the current approach for the final word selection in the (re)lexicalization methods is very naive, as the word to be used is picked by random from the list of approved word. Improving this sampling seems to be a low hanging fruit for achieving better results with my (re)lexicalization methods. Furthermore, the filtering to narrow down this list of approved word should be improved, as the current naive approach was unable to distinguish distinct sets of fitting and unfitting words.

Secondly, as mentioned in Chapter 5, it seems that identifying some specific POS-tags for lexicalization, and some specific word classes for relexicalization, might help to place the identifiers for (re)lexicalization approach in the templates in a more intelligent way.

Thirdly, as has been mentioned multiple times, my example for low-resource language was Finnish, which has a very complex morphology. This complexity might have given raise to problems with low-resource variant of the (re)lexicalization methods. This motivates two future directions: 1) the low-resource variant of (re)lexicalization should be applied to a language with less complex morphology, and 2) the morphological analysis and regeneration should be refined for Finnish language to obtain better results.

Fourthly, while human evaluation remains gold standard for assessing overall quality, it would be worth while to find suitable automatic evaluation metrics, which correlate with human judges, for fast and easily repeatable evaluation to aid the further research. Another aspect for future research regarding evaluation would be to evaluate the results on document level rather than on sentence level as was done in this thesis work.

Fifthly, an interesting way to examine, whether the results from my low-resource language approach are comparable in quality with those obtained with my high-resource, could be to use some high-resource language as the low-resource language in my round-trip approach. Obtaining results of the same level with both approaches would suggest that the round-trip proposed in this thesis does not compromise the output quality.

And as a final note, for the low- and high-resource languages used in my experiments, there were directly aligned cross-lingual embeddings available. However, this might be the case quite seldom. During the thesis work, I also conducted initial testing with

an additional language in between. In other words, I first retrieved the most similar word for a low-resource language from another low-resource language, and then finally the closest high-resource language word for the intermediate low-resource language word. These initial experiments indicate that this method might be useful and deserves future attention. The motivation was to confirm that my round-trip approach might be useful even without directly aligned word embeddings.

# Bibliography


[1] M. Artetxe, G. Labaka, and E. Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[2] H. Banaee, M. U. Ahmed, and A. Loutfi. Towards NLG for Physiological Data Monitoring with Body Area Networks. In *14th European Workshop on Natural Language Generation, Sofia, Bulgaria, August 8-9, 2013*, pages 193–197, 2013.

[3] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching Word Vectors with Subword Information. *CoRR*, abs/1607.04606, 2016.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. 2020.

[6] D. L. Chen and R. J. Mooney. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on Machine learning*, pages 128–135, 2008.

[7] T.-C. Chi and Y.-N. Chen. CLUSE: Cross-Lingual Unsupervised Sense Embeddings. *arXiv preprint arXiv:1809.05694*, 2018.

[8] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] L. Duong, H. Kanayama, T. Ma, S. Bird, and T. Cohn. Multilingual training of crosslingual word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 894–904, 2017.

[11] T. C. Ferreira, C. van der Lee, E. van Miltenburg, and E. Krahmer. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. *arXiv preprint arXiv:1908.09022*, 2019.

[12] J. R. Firth. *Papers in Linguistics, 1934-1951.* London, 1957.

[13] K. Fort, G. Adda, and K. B. Cohen. Amazon mechanical turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413–420, 2011.

[14] A. Gatt and E. Krahmer. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61, 03 2017.

[15] D. Gkatzia and S. Mahamood. A snapshot of NLG evaluation practices 2005-2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 57–60, 2015.

[16] J. K. Goodman, C. E. Cryder, and A. Cheema. Data collection in a flat world: The strengths and weaknesses of Mechanical Turk samples. *Journal of Behavioral Decision Making*, 26(3):213–224, 2013.

[17] A. Graefe. Guide to automated journalism. 2016.

[18] M. Hämäläinen. UralicNLP: An NLP library for Uralic Languages. *Journal of Open Source Software*, 4(37):1345, 2019.

[19] H. F. Hastie and A. Belz. A Comparative Evaluation Methodology for NLG in Interactive Systems. In *LREC*, pages 4004–4011, 2014.

[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

[22] A. S. Law, Y. Freer, J. Hunter, R. H. Logie, N. McIntosh, and J. Quinn. A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit. *Journal of clinical monitoring and computing*, 19(3):183–194, 2005.

[23] L. Leppänen, M. Munezero, M. Granroth-Wilding, and H. Toivonen. Data-Driven News Generation for Automated Journalism. In *The 10th International Natural Language Generation conference, Proceedings of the Conference*, pages 188–197, United States, 9 2017. The Association for Computational Linguistics.

[24] O. Levy, A. Søgaard, and Y. Goldberg. A strong baseline for learning cross-lingual word embeddings from sentence alignments. *arXiv preprint arXiv:1608.05426*, 2016.

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[27] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[29] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[30] G. A. Miller. WordNet: A Lexical Database for (e)nglish. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[32] V. Plachouras, C. Smiley, H. Bretz, O. Taylor, J. L. Leidner, D. Song, and F. Schilder. Interacting with financial data using natural language. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1121–1124, 2016.

[33] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training, 2018.

[34] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

[35] A. Ramos-Soto, A. J. Bugarín, S. Barro, and J. Taboada. Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Transactions on Fuzzy Systems*, 23(1):44–57, 2014.

[36] J. Reisinger and R. J. Mooney. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics, 2010.

[37] E. Reiter and R. Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, 1997.

[38] E. Reiter, R. Robertson, and L. M. Osman. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1-2):41–58, 2003.

[39] S. Ruder, I. Vulić, and A. Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[41] S. Sirén-Heikel, L. Leppänen, C.-G. Lindén, and A. Bäck. Unboxing news automation: Exploring imagined affordances of automation in news journalism. *Nordic Journal of Media Studies*, 1(1):47–66, 2019.

[42] W. L. Taylor. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

[43] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1218–1227, 2014.

[44] M. Ulčar and M. Robnik-Šikonja. FinEst BERT and CroSloEngual BERT: less is more in multilingual models, 2020.

[45] S. Upadhyay, K.-W. Chang, M. Taddy, A. Kalai, and J. Zou. Beyond bilingual: Multi-sense word embeddings using multilingual context. *arXiv preprint arXiv:1706.08160*, 2017.

[46] C. van der Lee, A. Gatt, E. van Miltenburg, S. Wubben, and E. Krahmer. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan, Oct.–Nov. 2019. Association for Computational Linguistics.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need, 2017.

[48] P. Vougiouklis, E. Maddalena, J. Hare, and E. Simperl. How biased is your NLG evaluation? In *Proceedings of the 1st CrowdBias Workshop*, pages 1–3, July 2018.

[49] L. Wanner, H. Bosch, N. Bouayad-Agha, G. Casamayor, T. Ertl, D. Hilbring, L. Johansson, K. Karatzas, A. Karppinen, I. Kompatsiaris, et al. Getting the environmental information across: from the web to the user. *Expert Systems*, 32(3):405–432, 2015.

[50] X. Yuan, T. Wang, C. Gulcehre, A. Sordoni, P. Bachman, S. Subramanian, S. Zhang, and A. Trischler. Machine comprehension by text-to-text neural question generation. *arXiv preprint arXiv:1705.02012*, 2017.